

EEEEEEEEEEEEEEEE	RRRRRRRRRRRRR	FFFFFFFFFFFFFF
EEEEEEEEEEEEEEEE	RRRRRRRRRRRRR	FFFFFFFFFFFFFF
EEEEEEEEEEEEEEEE	RRRRRRRRRRRRR	FFFFFFFFFFFFFF
EEE	RRR	FFF
EEEEEEEEEEEEEE	RRRRRRRRRRRRR	FFFFFFFFFFFFFF
EEEEEEEEEEEEEE	RRRRRRRRRRRRR	FFFFFFFFFFFFFF
EEEEEEEEEEEEEE	RRRRRRRRRRRRR	FFFFFFFFFFFFFF
EEE	RRR	FFF
EEEEEEEEEEEEEE	RRR	FFF
EEEEEEEEEEEEEE	RRR	FFF
EEEEEEEEEEEEEE	RRR	FFF

\*\*FILE\*\* ID\*\*UBAINT

K 14

UU UU BBBBBBBBBB AAAAAAA I II III NN NN TTTTTTTTTT  
UU UU BBBBBBBBBB AAAAAAA I II III NN NN TTTTTTTTTT  
UU UU BB BB AA AA I I NN NN TT  
UU UU BB BB AA AA I I NN NN TT  
UU UU BB BB AA AA I I NNNN NN TT  
UU UU BB BB AA AA I I NNNN NN TT  
UU UU BBBBBBBBBB AA AA I I NN NN NN TT  
UU UU BBBBBBBBBB AA AA I I NN NN NN TT  
UU UU BB BB AAAAAAAA I I NN NNNN TT  
UU UU BB BB AAAAAAAA I I NN NNNN TT  
UU UU BB BB AA AA I I NN NN TT  
UU UU BB BB AA AA I I NN NN TT  
UUUUUUUUUUUU BBBBBBBBBB AA AA I II III NN NN TT  
UUUUUUUUUUUU BBBBBBBBBB AA AA I II III NN NN TT

The diagram illustrates a 2D convolution operation. The input layer (left) consists of 10 'L' symbols arranged in a 10x1 column. The filter (top) consists of 5 'I' symbols arranged in a 5x1 column. The output layer (right) consists of 5 'S' symbols arranged in a 5x1 column. The result of the convolution is a 5x5 matrix of 'S' symbols, with the first row and column of the input 'L' symbols being zeroed out.

0001  
0002 C Version: 'V04-000'  
0003 C  
0004 C\*\*\*\*\*  
0005 C\*  
0006 C\* COPYRIGHT (c) 1978, 1980, 1982, 1984 BY  
0007 C\* DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.  
0008 C\* ALL RIGHTS RESERVED.  
0009 C\*  
0010 C\* THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED  
0011 C\* ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE  
0012 C\* INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER  
0013 C\* COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY  
0014 C\* OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY  
0015 C\* TRANSFERRED.  
0016 C\*  
0017 C\* THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE  
0018 C\* AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT  
0019 C\* CORPORATION.  
0020 C\*  
0021 C\* DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS  
0022 C\* SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.  
0023 C\*  
0024 C\*  
0025 C\*\*\*\*\*  
0026 C  
0027 C  
0028 C  
0029 C AUTHOR BRIAN PORTER CREATION DATE 3-FEB-1979  
0030 C  
0031 C Functional description:  
0032 C  
0033 C This module is called to display UBA interrupts.  
0034 C  
0035 C Modified by:  
0036 C  
0037 C V03-004 SAR0160 Sharon A. Reynolds, 13-Oct-1983  
0038 C Added an SYE update that makes the register heralds  
0039 C generic.  
0040 C  
0041 C V03-003 SAR0103 Sharon A. Reynolds, 20-Jun-1983  
0042 C Changed the carriage control in the 'format' statements  
0043 C for use with ERF.  
0044 C  
0045 C V03-002 SAR0054 Sharon A. Reynolds, 13-Jun-1983  
0046 C Removed brief/cryptic support.  
0047 C  
0048 C v03-001 BP0001 Brian Porter, 20-AUG-1982  
0049 C Minor edit.  
0050 C\*\*  
0051 C  
0052 C Subroutine UBA\_INTERRUPTS (lun,option)  
0053 C include 'src\$:msghdr.for /nolist'  
0113 C  
0114 C  
0115 byte lun

```

0116
0117      character*1      option
0118
0119
0120
0121      if (
0122      1 lib$extzv(24,8,emb$l_hd_sid) .eq. 255      ! 11/780
0123      1 or
0124      1 lib$extzv(24,8,emb$l_hd_sid) .eq. 1      ! 11/780's
0125      1 or
0126      1 lib$extzv(24,8,emb$l_hd_sid) .eq. 4      ! 11/7XX
0127      1 ) then
0128
0129      if (option .eq. 'S') then
0130
0131      call dw780_interrupt (lun)
0132      endif
0133
0134      c
0135      c      for additional UBA support the ELSE-IF-THEN should be expanded
0136      c      at this point.
0137      c
0138
0139      endif
0140
0141      return
0142      end

```

## PROGRAM SECTIONS

Name	Bytes	Attributes
0 \$CODE	80	PIC CON REL LCL SHR EXE RD NOWRT LONG
1 \$PDATA	8	PIC CON REL LCL SHR NOEXE RD NOWRT LONG
2 \$LOCAL	32	PIC CON REL LCL NOSHR NOEXE RD WRT LONG
3 EMB	512	PIC OVR REL GBL SHR NOEXE RD WRT LONG
Total Space Allocated	632	

## ENTRY POINTS

Address	Type	Name
0-00000000		UBA_INTERRUPTS

## VARIABLES

Address	Type	Name	Address	Type	Name
3-00000000	I*4	EMBSL HD SID	3-00000004	I*2	EMBSW HD ENTRY
3-0000000E	I*2	EMBSW HD ERRESEQ	AP-00000004a	L*1	LUN
AP-00000008a	CHAR	OPTION			

UBA\_INTERRUPTS

N 14  
16-Sep-1984 00:17:44  
5-Sep-1984 14:24:33

VAX-11 FORTRAN V3.4-56  
DISK\$VMSMASTER:[ERF.SRC]UBAINT.FOR;1

Page 3

ARRAYS

Address	Type	Name	Bytes	Dimensions
3-00000000	L*1	EMB	512	(0:511)
3-00000006	I*4	EMB\$Q_HD_TIME	8	(2)

FUNCTIONS AND SUBROUTINES REFERENCED

Type	Name	Type	Name
	DW780_INTERRUPT	I*4	LIB\$EXTZV

0001  
0002  
0003  
0004 SUBROUTINE DW780\_INTERRUPT (LUN)  
0005  
0006  
0007 INCLUDE 'SRC\$:MSGHDR.FOR /NOLIST'  
0008  
0009  
0010  
0011  
0012  
0013  
0014  
0015  
0016  
0017  
0018  
0019  
0020  
0021  
0022  
0023  
0024  
0025  
0026  
0027  
0028  
0029  
0030  
0031  
0032  
0033  
0034  
0035  
0036  
0037  
0038  
0039  
0040  
0041  
0042  
0043  
0044  
0045  
0046  
0047  
0048  
0049  
0050  
0051  
0052  
0053  
0054  
0055  
0056  
0057  
0058  
0059  
0060  
0061  
0062  
0063  
0064  
0065  
0066  
0067  
0068  
0069      BYTE      LUN  
0070  
0071      INTEGER\*4      FIELD  
0072      INTEGER\*4      COMPRESS4  
0073      INTEGER\*4      UBA\_CSR  
0074      INTEGER\*4      UBA\_CR  
0075      INTEGER\*4      UBA\_SR  
0076      INTEGER\*4      UBA\_DCR  
0077      INTEGER\*4      UBA\_FMER  
0078      INTEGER\*4      UBA\_FUBAR  
0079      INTEGER\*4      ADAPTER\_TR  
0080      INTEGER\*4      ERROR\_PC  
0081      INTEGER\*4      ERROR\_PSL  
0082  
0083      logical\*1      diagnostic\_mode  
0084  
0085      EQUIVALENCE      (EMB(16),UBA\_CSR)  
0086      EQUIVALENCE      (EMB(20),UBA\_CR)  
0087      EQUIVALENCE      (EMB(24),UBA\_SR)  
0088      EQUIVALENCE      (EMB(28),UBA\_DCR)  
0089      EQUIVALENCE      (EMB(32),UBA\_FMER)  
0090      EQUIVALENCE      (EMB(36),UBA\_FUBAR)  
0091      EQUIVALENCE      (EMB(40),ADAPTER\_TR)  
0092      EQUIVALENCE      (EMB(44),ERROR\_PC)  
0093      EQUIVALENCE      (EMB(48),ERROR\_PSL)  
0094  
0095      CHARACTER\*31      V1UBACR(0:6)  
0096      DATA      V1UBACR(0)      //'ADAPTER INIT'//  
0097      DATA      V1UBACR(1)      //'UNIBUS POWER FAIL'//  
0098      DATA      V1UBACR(2)      //'CONFIGURATION INTERRUPT ENABLE'//  
0099      DATA      V1UBACR(3)      //'SBI TO UNIBUS ERROR INT ENABLE'//  
0100      DATA      V1UBACR(4)      //'UNIBUS TO SBI ERROR INT ENABLE'//  
0101      DATA      V1UBACR(5)      //'BR INTERRUPT ENABLE'//  
0102      DATA      V1UBACR(6)      //'INTERRUPT FIELD SWITCH'//  
0103  
0104      CHARACTER\*25      V1UBASR(0:10)  
0105      DATA      V1UBASR(0)      //'UNIBUS ''SSYN'' TIMEOUT'//  
0106      DATA      V1UBASR(1)      //'UNIBUS SELECT TIMEOUT'//  
0107      DATA      V1UBASR(2)      //'LOST ERROR'//  
0108      DATA      V1UBASR(3)      //'MAP REGISTER PARITY FAIL'//  
0109      DATA      V1UBASR(4)      //'INVALID MAP REGISTER'//  
0110      DATA      V1UBASR(5)      //'DATA PATH PARITY ERROR'//  
0111      DATA      V1UBASR(6)      //'COMMAND TRANSMIT TIMEOUT'//  
0112      DATA      V1UBASR(7)      //'COMMAND TRANSMIT ERROR'//  
0113      DATA      V1UBASR(8)      //'CORRECTED READ DATA'//  
0114      DATA      V1UBASR(9)      //'READ DATA SUBSTITUTE'//  
0115      DATA      V1UBASR(10)      //'READ DATA TIMEOUT'//

```
0116  
0117      CHARACTER*3      TR  
0118      DATA   TR          /'TR '/  
0119  
0120  
0121      CALL FRCTOF (LUN)  
0122  
0123      call header (lun)  
0124  
0125      call logger (lun,"DW" INTERRUPT)  
0126  
0127      diagnostic_mode = .false.  
0128  
0129      if (lib$extzv(28,3,uba_dcr) .ne. 0) diagnostic_mode = .true.  
0130  
0131      CALL LINCHK (LUN,2)  
0132  
0133      WRITE(LUN,10) ERROR_PC  
0134      FORMAT(/' ',T8,'ERROR PC',T24,Z8.8)  
0135  
0136      if (.not. diagnostic_mode) then  
0137  
0138      CALL VAXPSL (LUN,ERROR_PSL)  
0139  
0140      CALL LINCHK (LUN,2)  
0141  
0142      WRITE(LUN,12) ADAPTER_TR  
0143      FORMAT(/' ',T8,'ADAPTER TR# ',I<COMPRESS4 (ADAPTER_TR)>,'.')  
0144  
0145      CALL UBA_REGA (LUN,UBA_CSR)  
0146      else  
0147  
0148      call linchk (lun,6)  
0149  
0150      write(lun,13) error_psl,adapter_tr,uba_csr  
0151      format(/' ',t8,'ERROR PSL',t24,z8.8,/,  
0152      1 t8,'ADAPTER TR# ',i<compress4 (adapter_tr)>,'.',/,  
0153      1 t8,'"DW" CSR',t24,z8.8)  
0154      endif  
0155  
0156      CALL LINCHK (LUN,1)  
0157  
0158      WRITE(LUN,15) UBA_CR  
0159      FORMAT(' ',T8,'"DW" CR',T24,Z8.8)  
0160  
0161      if (.not. diagnostic_mode) then  
0162  
0163      CALL OUTPUT (LUN,UBA_CR,V1UBACR,0,0,6,'0')  
0164  
0165      FIELD = LIB$EXTZV(26,5,UBA_CR)  
0166  
0167      IF (FIELD .NE. 0) THEN  
0168  
0169      FIELD = (FIELD*16) - 1  
0170  
0171      CALL LINCHK (LUN,1)  
0172
```

```
0173      20      WRITE(LUN,20) FIELD
0174      FORMAT(' ',T40,'MAPS 0. THRU ',I<COMPRESS4 (FIELD)>,'. DISABLED')
0175      ENDIF
0176      endif
0177
0178      CALL LINCHK (LUN,1)
0179
0180      25      WRITE(LUN,25) UBA_SR
0181      FORMAT(' ',T8,'''DW'' SR',T24,Z8.8)
0182
0183      if (.not. diagnostic_mode) then
0184
0185      CALL OUTPUT (LUN,UBA_SR,V1UBASR,0,0,10,'0')
0186
0187      DO 35,I = 24,27
0188
0189      IF (JIAND(UBA_SR,2**I) .NE. 0) THEN
0190
0191      CALL LINCHK (LUN,1)
0192
0193      30      WRITE(LUN,30) I - 20
0194      FORMAT(' ',T40,'BRRVR ',I1,'. FULL')
0195      ENDIF
0196
0197      35      CONTINUE
0198      endif
0199
0200      CALL LINCHK (LUN,1)
0201
0202      40      WRITE(LUN,40) UBA_DCR
0203      FORMAT(' ',T8,'''DW'' DCR',T24,Z8.8)
0204
0205      if (diagnostic_mode) then
0206
0207      CALL LINCHK (LUN,1)
0208
0209      45      WRITE(LUN,45)
0210      FORMAT(' ',T40,'DIAGNOSTIC MODE')
0211      endif
0212
0213      CALL LINCHK (LUN,1)
0214
0215      50      WRITE(LUN,50) UBA_FMER
0216      FORMAT(' ',T8,'''DW'' FMER',T24,Z8.8)
0217
0218      if (.not. diagnostic_mode) then
0219
0220      IF (JIAND(UBA_SR,'000006F8'X) .NE. 0) THEN
0221
0222      FIELD = LIBSEXTZV(0,9,UBA_FMER)
0223
0224      CALL LINCHK (LUN,1)
0225
0226      55      WRITE(LUN,55) FIELD
0227      FORMAT(' ',T40,'SELECTED MAP = ',I<COMPRESS4 (FIELD)>,'.')
0228      ENDIF
0229      endif
```

```

0230
0231      CALL LINCHK (LUN,1)
0232
0233      WRITE(LUN,60) UBA_FUBAR
0234      FORMAT(' ',T8,'"DW" FUBAR',T24,Z8.8)
0235
0236      if (.not. diagnostic_mode) then
0237
0238      IF (JIAND(UBA_SR,'00000003'X) .NE. 0) THEN
0239
0240      FIELD = JISHFT(JIAND(UBA_FUBAR,'0000FFFE'X),-1)
0241
0242      CALL LINCHK (LUN,1)
0243
0244      WRITE(LUN,65) FIELD
0245      FORMAT(' ',T40,'UNIBUS ADDRESS = ',05.5,'X (OCTAL)')
0246      ENDIF
0247      endif
0248
0249      RETURN
0250      END

```

## PROGRAM SECTIONS

Name	Bytes	Attributes
0 \$CODE	941	PIC CON REL LCL SHR EXE RD NOWRT LONG
1 \$PDATA	441	PIC CON REL LCL SHR NOEXE RD NOWRT LONG
2 \$LOCAL	792	PIC CON REL LCL NOSHR NOEXE RD WRT LONG
3 EMB	512	PIC OVR REL GBL SHR NOEXE RD WRT LONG
Total Space Allocated	2686	

## ENTRY POINTS

Address	Type	Name
0-00000000		DW780_INTERRUPT

## VARIABLES

Address	Type	Name	Address	Type	Name
3-00000028	I*4	ADAPTER TR	2-000001EC	L*1	DIAGNOSTIC MODE
3-00000000	I*4	EMBSL HD SID	3-00000004	I*2	EMBSW HD ENTRY
3-0000000E	I*2	EMBSW HD ERRSEQ	3-0000002C	I*4	ERROR_PC
3-00000030	I*4	ERROR_PSC	2-000001F0	I*4	FIELD
2-000001F4	I*4	I	AP-00000004a	L*1	LUN
2-000001ED	CHAR	TR	3-00000014	I*4	UBA_CR
3-00000010	I*4	UBA_CSR	3-0000001C	I*4	UBA_DCR
3-00000020	I*4	UBA_FMER	3-00000024	I*4	UBA_FUBAR
3-00000018	I*4	UBA_SR			

## ARRAYS

Address	Type	Name	Bytes	Dimensions
3-00000000	L*1	EMB	512	(0:511)
3-00000006	I*4	EMBSQ HD_TIME	8	(2)
2-00000000	CHAR	V1UBACR	217	(0:6)
2-000000D9	CHAR	V1UBASR	275	(0:10)

## LABELS

Address	Label								
1-00000039	10'	1-0000004F	12'	1-0000006D	13'	1-000000B2	15'	1-000000C6	20'
1-00000101	30'	**	35'	1-00000119	40'	1-0000012E	45'	1-00000145	50'
1-0000017B	60'	1-00000192	65'					1-0000015B	55'

## FUNCTIONS AND SUBROUTINES REFERENCED

Type	Name	Type	Name	Type	Name	Type	Name	Type	Name	Type	Name
I*4	COMPRESS4					I*4	LIB\$EXTZV				
OUTPUT	UBA_REGA			HEADER	VAXPSL			LINCHK			LOGGER

## COMMAND QUALIFIERS

FORTRAN /LIS=LIS\$:UBAINT/OBJ=OBJ\$:UBAINT MSRC\$:UBAINT  
 /CHECK=(NOBOUNDS,OVERFLOW,NOUNDERFLOW)  
 /DEBUG=(NOSYMBOLS,TRACEBACK)  
 /STANDARD=(NOSYNTAX,NOSOURCE FORM)  
 /SHOW=(NOREPROCESSOR,NOINCLUDE,MAP)  
 /F77 /NOG\_FLOATING /I4 /OPTIMIZE /WARNINGS /NOD\_LINES /NOCROSS\_REFERENCE /NOMACHINE\_CODE /CONTINUATIONS=19

## COMPILATION STATISTICS

Run Time: 4.00 seconds  
 Elapsed Time: 10.42 seconds  
 Page Faults: 181  
 Dynamic Memory: 185 pages

0154 AH-BT13A-SE  
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION  
CONFIDENTIAL AND PROPRIETARY